COM

IUnknown::QueryInterface

Returns a pointer to a specified interface on an object to which a client currently holds an interface pointer. This function must call IUnknown::AddRef on the pointer it returns.

Copy Code

```
HRESULT QueryInterface(
   REFIID iid,
   void ** ppvObject
);
```

Parameters

iid

> [in] Identifier of the interface being requested.

ppvObject

> [out] Address of pointer variable that receives the interface pointer requested in *riid*. Upon successful return, *ppvObject* contains the requested interface pointer to the object. If the object does not support the interface specified in *iid*, *ppvObject* is set to NULL.

Return Value

S_OK if the interface is supported, E_NOINTERFACE if not.

Remarks

The QueryInterface method gives a client access to other interfaces on an object.

For any one object, a specific query for the IUnknown [http://msdn.microsoft.com/en-us/library/ms680509 (VS.85).aspx] interface on any of the object's interfaces must always return the same pointer value. This allows a client to determine whether two pointers point to the same component by calling QueryInterface on both and comparing the results. It is specifically not the case that queries for interfaces (even the same interface through the same pointer) must return the same pointer value.

There are four requirements for implementations of QueryInterface (In these cases, "must succeed" means "must succeed barring catastrophic failure."):

- The set of interfaces accessible on an object through IUnknown::QueryInterface must be static, not dynamic. This means that if a call to QueryInterface for a pointer to a specified interface succeeds the first time, it must succeed again, and if it fails the first time, it must fail on all subsequent queries.
- It must be reflexive â€" if a client holds a pointer to an interface on an object, and queries for that interface, the call must succeed.
- It must be symmetric â€" if a client holding a pointer to one interface queries successfully for another, a query through the obtained pointer for the first interface must succeed.
- It must be transitive â€" if a client holding a pointer to one interface queries successfully for a second, and through that pointer queries successfully for a third interface, a query for the first interface through the pointer for the third interface must succeed.

Notes to Implementers

Implementations of QueryInterface must never check ACLs. The main reason for this rule is because COM requires that an object supporting a particular interface always return success when queried for that interface. Another reason is that checking ACLs on QueryInterface does not provide any real security because any client who has access to a particular interface can hand it directly to another client without any calls back to the server. Also, because COM caches interface pointers, it does not call QueryInterface on the server every time a client does a query.

Requirements

For an explanation of the requirement values, see Requirements (COM) [http://msdn.microsoft.com/en-us/library/ms693432(VS.85).aspx] .

Windows NT/ 2000/ XP: Requires Windows NT 3.1 or later.

Windows 95/ 98: Requires Windows 95 or later.

Header: Declared in unknwn.h.

Send comments about this topic to Microsoft.

Tags:

Community Content

| Check out the COM introductive material | Last Edit 1.38 PM by JohnnyTH |

If you can't understand why the implementing method must call AddRef() on the returned pointer, check out the COM introduction which is very clear about this. (Up two levels, "Guide").

Tags: contentbug